Mérnöki programozás 2

Szerkesztette: dr. Vass Péter Tamás

Algoritmusok tervezése és leírása

1. feladat:

Adjuk össze a számokat 1-től 10-ig egy elöltesztelő ciklus segítségével és írjuk ki az eredményt!

s = 1+2+3+4+….+10

s = 1
s = s+2
s = s+3
….
s = s+10

A probléma elemzése:

- a tartomány alsó és felső határát két konstans érték határozza meg,
- a számított részösszegek tárolásához be kell vezetnünk egy változót (pl. s),
- a számok összegzéséhez egy ciklust kell alkalmaznunk,
- a ciklus blokkjában az aktuális részösszeghez a sorban következő számot hozzá kell adnunk, melynek értékét egy másik változóban tároljuk (pl. x),

Algoritmusok tervezése és leírása

A probléma elemzése:

- az x egy ciklus változó, melynek értékét minden ismétlődés után növelni kell eggyel,
- a ciklus ismétlődését meghatározó feltétel az x aktuális értékére
- vonatkozik, amely nem lehet nagyobb 10-nél (x <= 10)

Algoritmusok tervezése és leírása

Folyamatábra készítése yED folyamatábra készítő szoftverrel

- rácsozat beállítása a szerkesztő ablakban: View → Grid vagy az eszköztár Grid gombja,
- rácsozat típusának és sűrűségének beállítása: File → Preferences → Grid,
- az algoritmusok folyamatábráinak elemei: Palette ablak → Flowchart
- új alakzat létrehozása a szerkesztő ablakban: bal egérgombbal behúzás a Palette ablakból,
- alakzat kiválasztása a szerkesztő ablakban: egér kattintással,
- kiválasztott alakzat törlése: Del gomb lenyomása, vagy Edit → Delete, vagy Delete gomb az eszköztáron,
- kiválasztott alakzat tulajdonságainak módosítása: a Properties View ablakban, vagy jobb egérgombbal a kiválasztott alakzat felett aktivált helyi menüből a Properties menüpont kiválasztása

Algoritmusok tervezése és leírása

Folyamatábra készítése yED folyamatábra készítő szoftverrel

- legfontosabb tulajdonságok: Text (az alakzatba foglalt szöveg), Fill Color, Fill Color 2 (alakzat kitöltési színei, a két szín az árnyékoláshoz kell), Line Color (az alakzat keretvonalának színe), Line Type (a keretvonal típusa), X és Y (az alakzat bal felső sarkának koordinátái), Width és Height (az alakzat szélessége és magassága),
- a kiválasztott alakzat pozíciója és mérete az egér segítségével is módosítható (vonszolás, ill. az alakzat körüli fekete négyzetek húzása),
- a szerkesztő ablakban lévő ablakok Ctrl+C és Ctrl+V technikával másolhatók,
- alakzatok összekötése vonallal: a kiindulási alakzat felett lenyomva tartott bal egérgombbal elindulva, a cél alakzatig kell húzni az egeret (egyik alakzatot sem szabad kijelölni kattintással a művelet előtt)

Algoritmusok tervezése és leírása

Folyamatábra készítése yED folyamatábra készítő szoftverrel

- az összekötő vonal helyzete utólag egérrel változtatható,
- a kiválasztott összekötő vonal tulajdonságainak beállítása: Properties View ablakban
- legfontosabb vonal tulajdonságok: Line Color (vonal szín), Line Type (vonal típus), Source Arrow (a kiindulási alakzat felé mutató nyíl típusa), Target Arrow (a cél alakzat felé mutató nyíl típusa)
- az összekötő vonalba töréspontok is létrehozhatók, ha a kívánt pontban az egérgombbal kattintunk, és utána tovább húzzuk a vonalat az egérrel,
- derékszögű megtöréshez be kell kapcsolni az Orthogonal Edges funkciót: View → Orthogonal Edges, vagy az eszköztár megfelelő gombja

Algoritmusok tervezése és leírása

Folyamatábra készítése yED folyamatábra készítő szoftverrel

- az összekötő vonalhoz szöveges címke is rendelhető: Text tulajdonság
- a címke helyzete és tulajdonságai is állíthatók
- a címke helyzetének szabad változtatását biztosító beállítás: Properties View ablak → Placement tulajdonság → Modell: Free és Position: Anywhere,
- ábra mentése: File → Save, vagy az eszköztár megfelelő gombja,
- egyéb hasznos funkciók: nagyítás, kicsinyítés (View menü), váltás a navigációs és szerkesztési üzemmódok között (Edit Mode / Navigation Mode az eszköztáron)
- a Structure View ablakban láthatók az ábrát alkotó grafikus elemek, itt is ki lehet választani a rajzi objektumokat, más alakzat által lefedett alakzat kijelölésére is alkalmas,

Algoritmusok tervezése és leírása

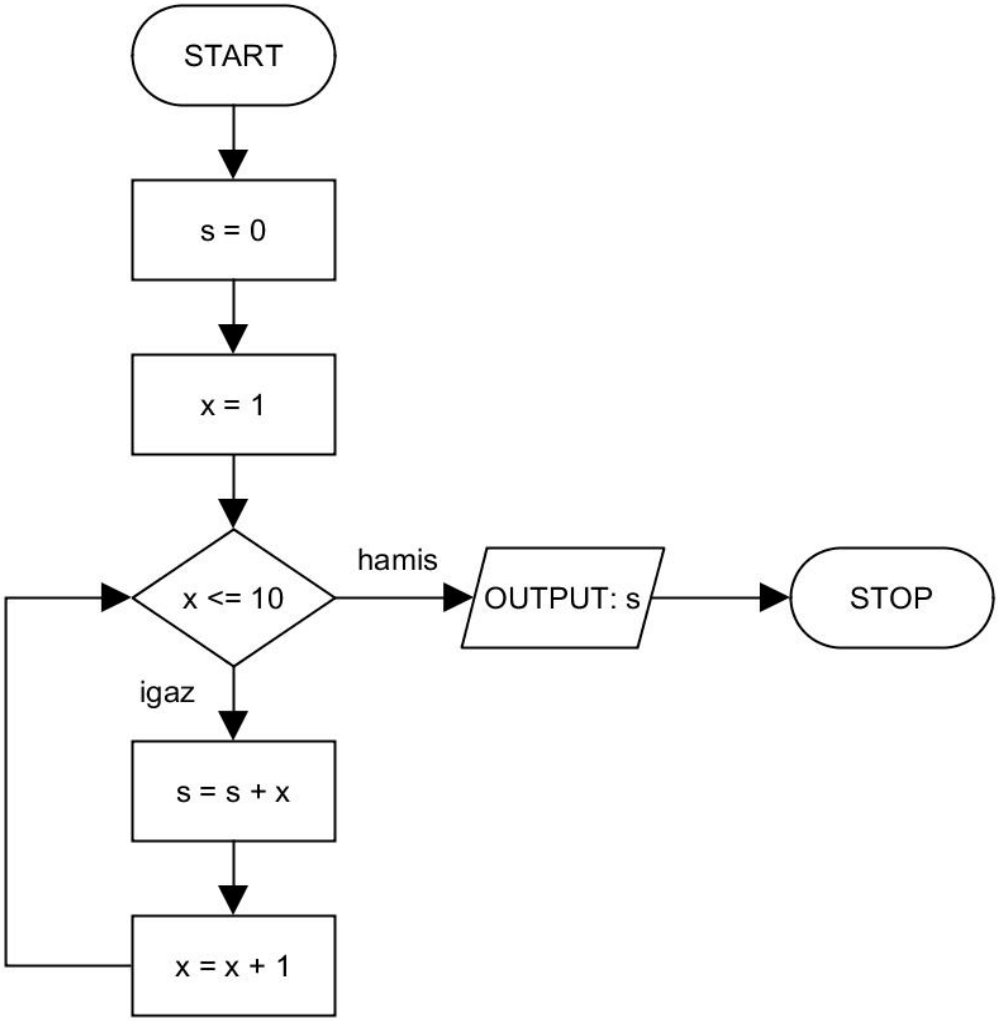
Folyamatábra készítése yED folyamatábra készítő szoftverrel

- a Neighbourhood ablakban az éppen kiválasztott rajzi elem szomszédsága jelenik meg,
- az Overview ablak egy áttekintő képet ad a rajzról,
- egyszerre több rajzi elem is kiválasztható és csoportba rendezhető (Shift + kattintás, vagy a befoglaló terület bekeretezése egérrel és Grouping → Group)
- a csoportba foglalt elemek egyszerre mozgathatók,
- kiválasztott csoport felbontása: Grouping → Ungroup
- elkészült rajz exportálása: File → Export → mentés helye, fájl neve és típusa → Scaling factor beállítása

Algoritmusok tervezése és leírása

Az algoritmus pszeudokódja és folyamatábrája:

```
s = 0
x = 1
while x <= 10 do
        s = s + x
        x = x + 1
write s
```



Algoritmusok tervezése és leírása

2. feladat:
Szorozzuk össze a páros számokat 1-től 10-ig egy hátultesztelő ciklus segítségével és írjuk ki az eredményt!

3. feladat:
Számítsuk ki az $s = \sum_{i=1}^{n} (-1)^i \cdot \frac{i-1}{i+1}$ összeget, és írjuk ki az eredmény!

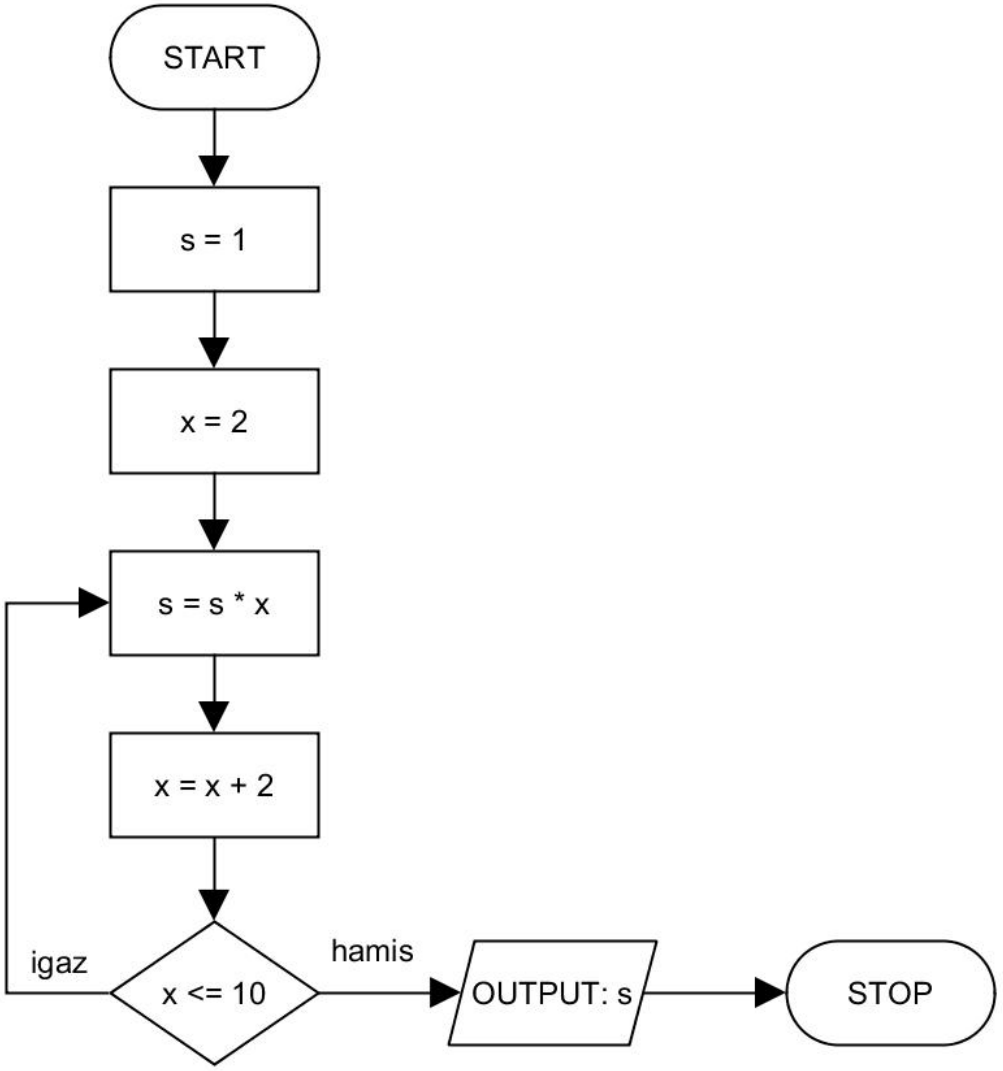
4. feladat:
Számítsuk ki a $q = \prod_{i=1}^{n} \frac{i}{i+1}$ szorzatot, és írjuk ki az eredményt!

5. feladat:
Olvassunk be egy tetszőleges x valós számot, vizsgáljuk meg, hogy pozitív, negatív, vagy nulla! A vizsgálat eredményét írjuk ki!

Algoritmusok tervezése és leírása

2. feladat folyamatábrája és pszeudokódja:

```
s = 1
x = 2
do
   s = s * x
   x = x + 2
while x <= 10
write s
```



Algoritmusok tervezése és leírása

2. feladat algoritmusának táblázatos kiértékelése:

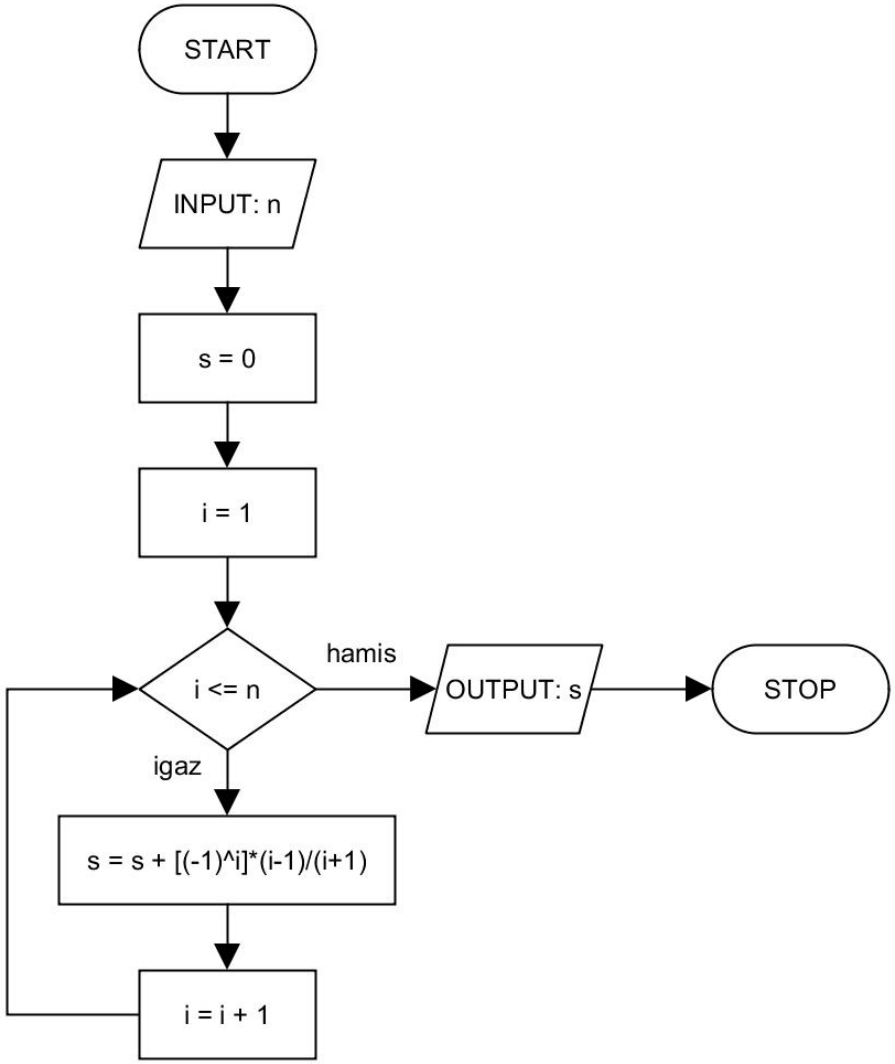
ismétlések száma	s értéke	x értéke
0	1	2
1	2	4
2	8	6
3	48	8
4	384	10
5	3840	12

Algoritmusok tervezése és leírása

3. feladat folyamatábrája és pszeudokódja:

```
read n
s = 0
i = 1
while i <= n do
    s = s + [(-1)^ i ]*( i-1)/( i+1)
    i = i + 1
write s
```

```
read n
s = 0
for i=1 to n do
    s = s + [(-1)^i ]*( i-1)/( i+1)
write s
```

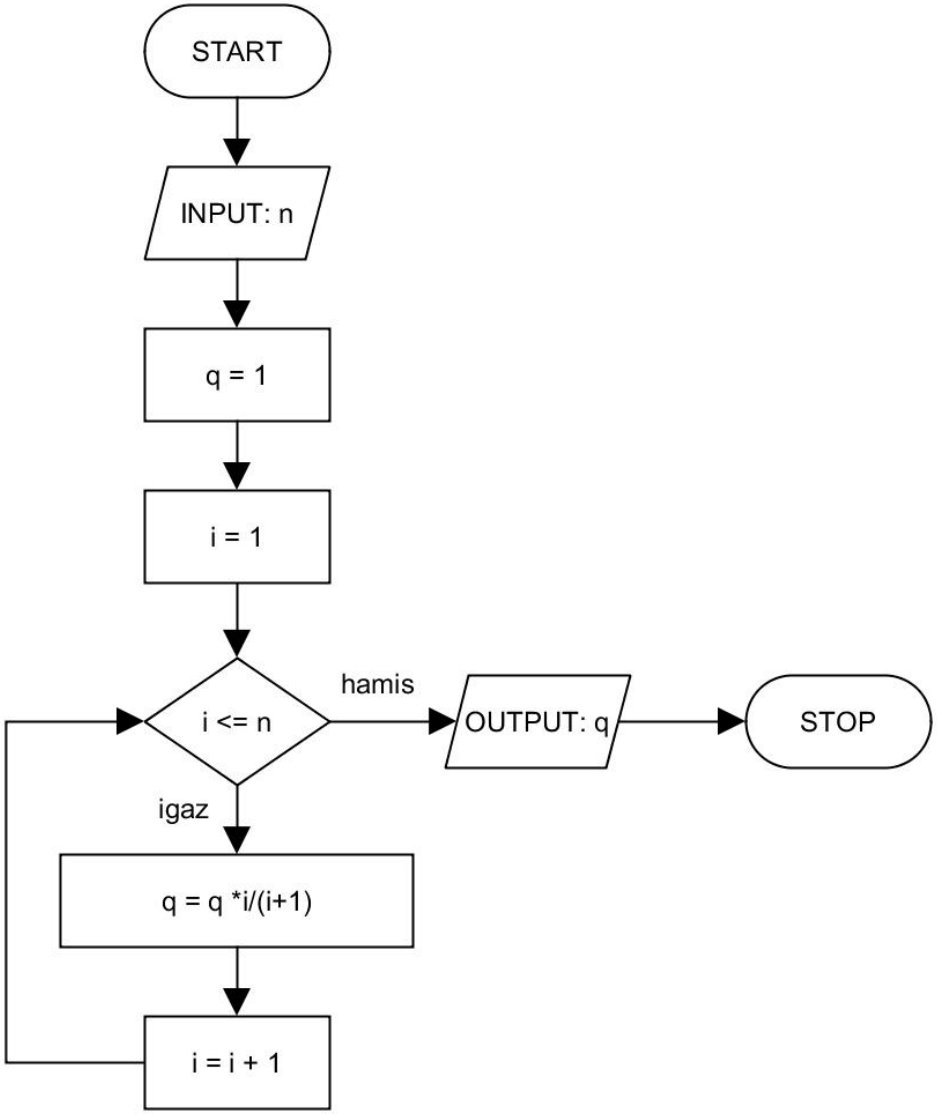


Algoritmusok tervezése és leírása

4. feladat folyamatábrája és pszeudokódja:

```
read n
q = 1
i = 1
while i <= n do
   q = q * i /( i+1)
   i = i + 1
write q
```

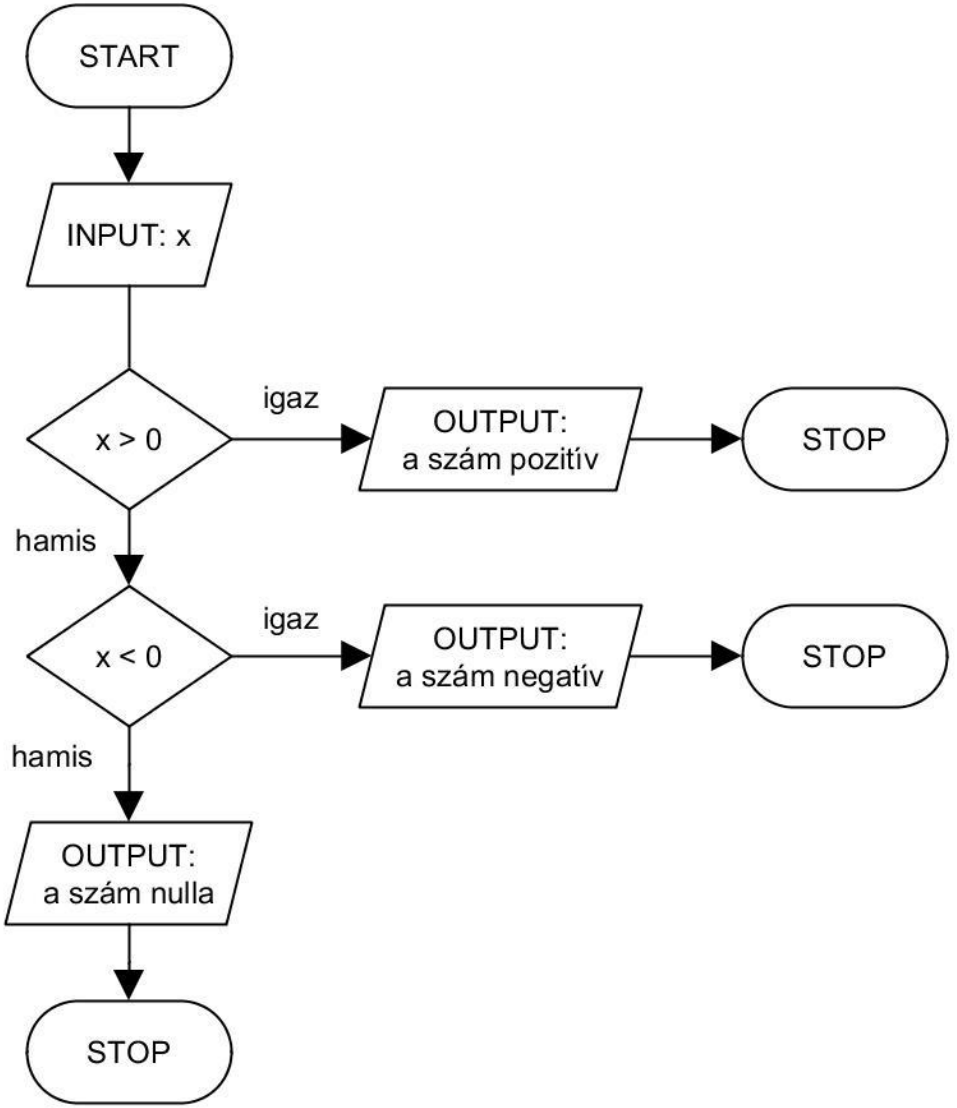
```
read n
q = 1
for i=1 to n do
   q = q * i /( i+1)
write q
```



Algoritmusok tervezése és leírása

5. feladat folyamatábrája és pszeudokódja:

```
read x
if x > 0 then
   write „a szám pozitív”
else if x < 0 then
   write „a szám negatív”
else
   write „a szám nulla”
```



Algoritmusok implementálása C programozási nyelven

1. feladat algoritmusának C forráskódja (elöltesztelő ciklus):

```
#include <stdio.h>

main(){
    int s,  x;

    s = 0;
    x = 1;
    while (x <= 10){
        s = s + x;
        x = x + 1;
    }
    printf("Az osszeg: %d", s);
}
```

Algoritmusok implementálása C programozási nyelven

1. feladat algoritmusának C forráskódja (hátultesztelő ciklus):

```
#include <stdio.h>
main(){
    int s, x;

    s = 0;
    x = 1;
    do{
        s = s + x;
        x = x + 1;
    } while (x <= 10);
    printf("Az osszeg: %d", s);
}
```